91602
new 82-10

COMPLIMENTARY COPY
Please Credit Ontario Hydro Research if
Reproduced in Whole or in Part

TO: Dr. D. Mills
Director of Research

ontario hydro
research division

COMPUTER VISION FOR ROBOTIC WELDING

David L. Mader

A demonstration has been made of a prototype computer vision system for robotic welding. A sheet of light was projected onto two metal plates to be welded together at a machined V-groove joint. The resulting stripe of light seen where the sheet of light intersected the surface of the plates was captured by a video camera viewing from an angle so that the stripe was seen to follow the contours of the surface. A frame grabber board plugged into a microcomputer provided an interface to the analog video signal. Software was written to control the frame grabber and to analyze the images and find the coordinates of key points in the area to be welded.

INTRODUCTION

As an essential component of an automated pipe welding system, a sensor system is used to detect the local values of position (centre of the weld groove and location of the groove walls) and the geometry of the weld joint (root gap, high/low misfit, and joint volume.) This information, measured in real time, is sent to the controller for interactive feedback control of welding torch position and welding parameters. Among various possible approaches, a computer vision technique has been investigated in the present work for the development of such a sensor system.

Fig 1. A computer vision system found the shoulders and edges of two plates to be welded together even in the presence of spatter.

A simple computer vision system has been demonstrated for the case of two metal plates to be butt welded together. The front end of the system was an optical triangulation scheme in which a sheet of light was projected

job	file	date	report no.
16-665-050	819.36	October 6, 1986	86-229-K

vertically onto the two pieces to be welded together, in such fashion that a stripe of light was seen on the workpiece at right angles to the seam. Since the plates were prepared by machining a bevel on the edges to be joined, the "joint preparation" was a V-groove. With the video camera arranged to observe the samples from an angle about 45 degrees different from the direction of projection of the sheet of light, the V-groove could be seen in the camera view. The joint geometry seen in cross section is given in Fig 2.

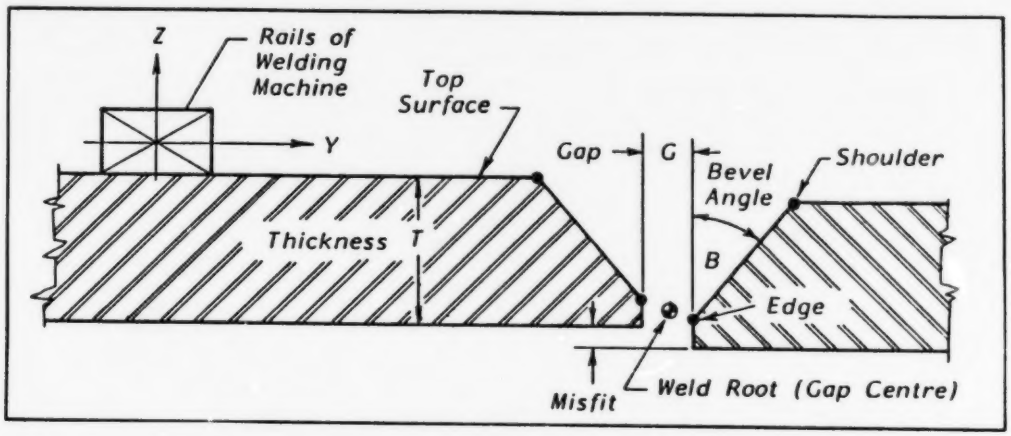


Fig 2. Joint geometry to be measured prior to the root pass weld consists of the gap and misfit, along with the location of the centre of the gap.

The gap width can be computed as the difference between the horizontal coordinates of the end points seen in Fig 1, and the gap centreline would be their average. Similarly, the vertical coordinates would yield the misfit and the height of the gap. A calculation needed for the geometry of viewing to convert from vertical dimensions on Fig 1 to height at the workpiece is discussed later.

In the literature, vision systems are espoused for several distinct tasks in pipe welding:

1. Guidance of the welding torch for single pass and multipass welding
2. Provision of joint geometry data for interactive feedback control of welding parameters.
3. Inspection of individual weld passes.
4. Inspection of the completed weld.

A system that could assist pipe welding with the above tasks could also be of value in repair welding to fill cavities gouged out at damage sites or in application of protective alloys by weld deposition.

LITERATURE SURVEY ON COMPUTER VISION FOR WELDING

An extensive literature survey on automated pipe welding has been done by Grant et al/1/, in which several papers on vision systems are identified.

Structured lighting generally appears to be the key to industrially successful vision systems, wherein either a narrow beam or a sheet of light is projected on to the object. Then the object is viewed by a TV camera, or a line camera which detects one TV line. With the simple line cameras, a separate means is used for scanning.

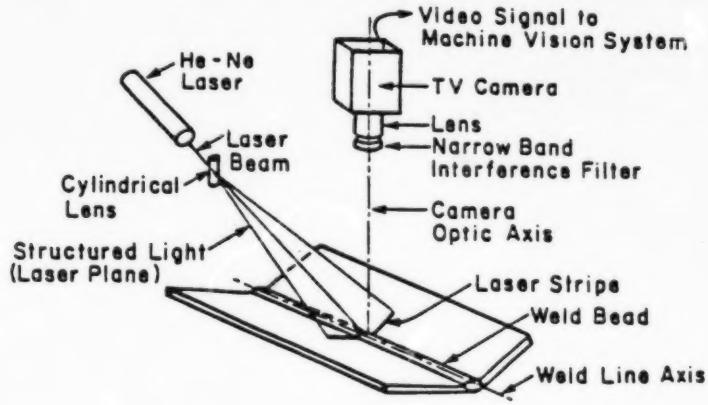


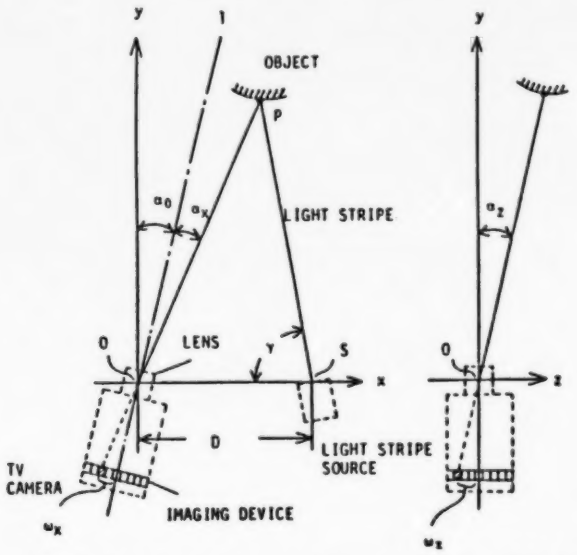
Fig 3. Light sectioning technique to obtain depth information.
Source: Agapakis et al/3/.

Many system tradeoffs can be made in a search for a winning combination. These tradeoffs affect cost, analysis time, and accuracy of results. In hardware tradeoffs, consideration can be given to use of off-the-shelf hardware such as commercially available frame grabbers, and especially mass produced items such as video cameras and personal computers to reduce initial procurement costs. To reduce analysis time, recourse can be taken to high speed computers, and custom-built hardware such as video preprocessor boards. System costs will also rise for specialized optical concepts such as scanners to move a beam of light across the surface.

Software tradeoffs include a balance between purchased packages of image processing routines, for example, and in-house development of software. Another choice is between high level languages to reduce the cost of software development, versus assembly language for greater speed of execution. The algorithms may be simple for fast execution, or complex for sophisticated analysis.

Drews and Starke/2/ describe two systems for welding torch guidance. A simple one dimensional non-scanned sensor responded to reflectivity variations across the joint. While processing time was only 0.03 sec, this system does not provide joint topology. A more conventional system

with fan beam illumination and a two dimensional 128 by 128 pixel (picture element) sensor was able to provide the joint centre, gap width, joint volume, angle of joint preparation, and misalignment of edges. Through use of laser illumination and an interference filter in front of the camera to exclude ambient light, the detected image was free of clutter so that a simple weighting algorithm was able to locate the centreline of the stripe of light. The cycle time is quoted as 0.2 sec. (No information is given about the computer.) The authors consider this an order of magnitude too slow for process control and intend to turn to a high speed microcomputer with 16-bit multiprocessor architecture.



$$x = \frac{D \tan(\alpha_0 + \alpha_x)}{\cot \gamma + \tan(\alpha_0 + \alpha_x)} \quad \ldots$$

$$y = \frac{D}{\cot \gamma + \tan(\alpha_0 + \alpha_x)} \quad \ldots$$

$$z = \frac{D \tan \alpha_z}{\cot \gamma + \tan(\alpha_0 + \alpha_x)} \quad \ldots$$

Fig 4. Optical triangulation. The coordinates x,y,z of a point P on the object can be found from the system geometry and the detected position of the image of P on the TV camera. Source: Ozeki et al/4/.

Agapakis et al/3/ describe an inspection system and show its performance on a fillet weld between two plates inclined to each other at a right angle. Their optical system, shown above in Fig 3, projects a fan beam on to the surface and has a video camera to view it from a different angle. Formulas for optical triangulation given by Ozeki et al/4/ are shown in Fig 4. In their computer analysis, they tried algorithms from the discipline of computer vision as exemplified in the book by Ballard and Brown/5/. At the image processing stage, they first tried use of a matched filter for stripe extraction but found it was too slow. They then fell back to the more customary use of simple thresholding. To reduce the observed stripe of light to a line, they first rejected "stripes" of the wrong width. Gaps were filled with a simple raster drawing routine, then data was smoothed and noise rejected with a one dimensional median filter. A "hop-along" split and merge segmentation routine was used to approximate the

stripe centreline with elementary straight line segments. The next step, image understanding, was implemented by fitting straight line segments to the observed stripe centreline. The breakpoints between the fitted line segments were initially identified as locations were the second derivative of the stripe was large. These breakpoints gave the junction between the weld bead and the top surface of the plates. The extrapolated intersection of the top surfaces is taken to be the weld root. The crossectional area of the fillet was calculated. Defects such as undercuts (this was a weld inspection system) are taken to be the differences between the extracted stripe and the fitted straight line segments.

Processing times for the above algorithm on a 100 by 100 pixel window were 0.1 sec for image acquisition, 0.3 sec for stripe detection, and 0.5 sec for representation, recognition and modeling, leading to a cycle time of 0.9 sec. No information is given on the type of computer. The authors intend to seek increased speed by use of assembly language and multiple processors.

A specialized optical triangulation sensor head described by Begin et al/6/ has a one dimensional sensor, an illumination beam, and a dual purpose scanner which simultaneously aims the illumination direction and the viewing direction. Since the available light is left concentrated in a narrow beam, the spot where it intersects the surface is brighter. This is important for a vision system to be used while the welding arc is on, so that the vision system illumination can be detected while rejecting the arc light. The acquisition time for one range profile was quoted as 0.01 to 0.1 sec. No discussion was given of data analysis or of system performance.

A tracking control system that was used for welding penstocks is described by Kawahara and Matsui/7/. This system could be used in a one pass mode in which measurements were made of joint geometry a short distance ahead of the torch. The software was able to cull out image points due to spatter. Performance was an accuracy of 0.05 mm, and a cycle time of 0.17 sec with an Intel 8080 microprocessor chip. The optical head contained no moving parts, using fan beam illumination and a two dimensional sensor.
The excellent speed of the system was due in part to use of a small size 32 by 32 pixel sensor requiring that only 1024 data points be analyzed. In analysis, each row in the image was reduced to a set of five numbers, along the lines of run length encoding in binary image analysis. The goal of the guidance system was simply to find the centre of the V-groove to be welded. No information was obtained on gap or misfit. Accordingly the data from the top surfaces was cropped away. Similarly, to avoid any confusing signals from the bottom of the groove this area was excluded. What remained were line segments from the groove walls, to which two straight lines were fitted. The extrapolated intersection point of these lines gave the coordinates of the gap centre. (Note that the misfit must be zero for this method to find the gap centre exactly.) The precision of the final coordinates was considerably greater than that indicated by the pixel size, since several data points contributed to each fitted line.

The robustness of the system was not discussed. Since data points were sorted into left and right side sets according to the location of the gap centre found in the previous cycle, the system has the potential to go astray due to temporary disturbances.

COMPUTER VISION SYSTEM FOR DEMONSTRATION

The optical system of Fig 3 was used for the demonstration of this report. The illumination source and TV camera were borrowed from other projects for the demonstration and do not constitute the final system choices. Two illumination souces were used with similar results: a slide projector containing a black slide with a scratch across it, and a small helium-neon laser with a glass rod to spread the beam out into a fan beam. The camera was a video camera with a vidicon tube and a long focal length lens. The frame grabber, which makes the video images accessible to a computer, was an Oculus-200 board made by Coreco, and it was installed in a Compaq 286 microcomputer.

The frame grabber takes as input any video source with output in standard TV format, namely RS-170 interlaced composite video. In this format two fields are taken, each in 1/60 sec, to acquire one complete frame in 1/30 sec. Each horizontal line is sampled at 512 points, each sample being converted to a 7-bit number (yielding intensity resolution to 128 gray levels) and stored in a memory bank on the frame grabber board. A total of 480 lines are thereby digitized and stored for each frame. Since this is done in real time, it can be seen that the frame grabber's analog to digital converter is required to process 512 X 480 = 245,760 pixels in 1/30 sec. The grabber also has a composite video output for a monitor, which displays the contents of the on-board memory via an on-board digital to analog converter which scans the memory in 1/30 sec.

To the computer, the frame grabber is a peripheral device plugged into the computer's bus. As such, it is accessed as an absolute address. In BASIC the OUT statement issues commands to the frame grabber, while the INP function is used to retrieve data from it. Actually, the frame grabber has five on-board registers, each with its own absolute address. The STATUS register can be monitored to ascertain whether the frame grabber is busy in the middle of acquiring a frame, for example. This register is written into at the appropriate time to initiate acquisition of the next frame that is desired. When a frame has been acquired and thereby "frozen", the computer can obtain the intensity at a given pixel by writing the desired pixel coordinates to the X and Y registers and reading the contents of the DATA register. To quickly read out the pixel intensities along a horizontal, vertical, or diagonal line, an auto-incrementing mode is provided. The starting location is given to the X and Y registers and the desired direction of travel is given to the DIRECTION register. Then, one simply reads the DATA register once for each pixel desired along the specified line. The frame grabber board will automatically increment the X and Y registers by the appropriate amounts.

Graphics displays can be generated by the same processes used to read out pixel intensities from an acquired frame. For instance, to plot a point its coordinates are written in to the X and Y registers and its intensity is written into the DATA register. In this mode it is possible to input an 8-bit number, so the graphics mode can display any one of 256 gray levels. Thus, for example, it is possible to display a window and view it while images are being acquired.

Image intensities can be transformed, to "binary vision" form for example, through use of programmable look-up tables located between the input analog to digital converter and the on-board memory. Such transformations are done on-line and do not cost any time delays. For binary vision, the look-up table (LUT) is set with its output as zero whenever the input is less than 63, and is set to 127 otherwise. The corresponding image on the monitor appears to be a silhouette. For "normal vision" the LUT output is loaded to match the input.

In the BASIC computer program given in the Appendix, the first section up to line 1000 is devoted to control of the frame grabber board (except for the subroutine at line 500.) When the program is run in interpretive BASIC, the frame grab commands are issued too late to acquire a complete frame, leaving the top portion of the image uncertain. When compiled, operation is as expected.

To plan an algorithm for use in the welding problem, it is valuable to consider time information. From the literature survey above, it appears that the desired cycle time is under 0.2 sec, with one paper calling for 0.02 sec and another claiming a useful industrial system with a cycle time of 0.17 sec. Of course the lower limit on cycle time would be the time to acquire at least one field (either the even or the odd half frame) of 1/60 = 0.017 sec.

A large time investment is needed for the computer to handle all the pixels in a full size image. For example, simply to transfer the contents of the on-board frame grabber memory to computer memory takes 1.2 sec using an assembly language routine provided by the manufacturer of the frame grabber. This result immediately rules out any algorithm which processes a full image, pixel by pixel. One therefore turns either to processing a small portion of the image, or to searching through the full image along a small number of lines. Processing a portion of the image contained in a 100 X 100 pixel window would treat as much data as some of the papers in the literature survey and may be feasible since the transfer time would be less than 0.05 sec. The lens for the camera would be chosen so that the critical features of the joint would all lie within the window. The scene outside the window though not used by the computer could be useful for the welder to look at.

A FAST SEARCH ALGORITHM TO FIND FEATURE POINTS

The algorithm chosen for this demonstration was a search algorithm that relies upon the image to have a definite pattern. As such, the algorithm is heuristic, though this was not felt to be a disadvantage since the welding problem is relatively well defined. General image processing techniques were shunned because they require calculations at all pixels within the analysis window. In this report the full image was used to obtain maximum resolution with a simple algorithm. In future work, modeling or fitting could be done at critical points to obtain better resolution and this could be traded off for faster speed by working in a reduced size window.

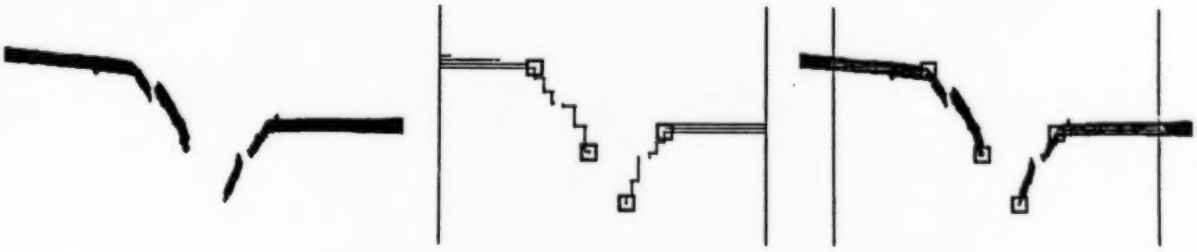
The problem that was assumed for this work was to find the shoulders and edges of two plates to be joined by a butt weld. It was assumed that there would be a definite gap between the two plates. The stripe of light seen on the plates due to the projected plane of light was assumed to be horizontal where it falls on the top surfaces and to extend to the edges of the image. The inclined lines of light from the V-groove joint preparation were assumed to slope downwards from the shoulder to the edge. The image then looked like two hockey sticks with the handles horizontal and the blades hanging down, with the ends of the blades nose to nose. This form for the image could always be obtained for a V-groove preparation through appropriate orientation of the projector and the camera.

The search algorithm seen in the Appendix starting at program line 1000 acquires a binary image and then finds the handles on the hockey sticks by scanning down a vertical search line near each side of the image and keeping track of all white dots (intensity in binary image > 63 .) Next horizontal lines are tracked from all dots found on the search lines and then the longest line is selected for each side. The end of the longest horizontal line is taken to be the shoulder of the V-groove and a small box is drawn there to indicate it. (Clearly better resolution could be obtained by fitting a model of the shoulder to the data in this area.) Then a descending staircase search is performed to find the end of the blade of the hockey stick, and another small box is drawn there to indicate the edge of the gap. If the stripe of light does not show up in the binary image in the groove region, the edges are found to coincide with the shoulders. If the stripe of light is seen in the groove and if there is no gap, the edge of the left plate will be found part way across the right plate and vice versa. If the gap is backlit, the gap region will be bright and the search will end up at the bottom of the screen. Clearly, backlighting is to be avoided, and improvement is needed for the zero gap case. However the algorithm performs well when it is presented with the assumed image described above.

To reduce search time when the image is a wide stripe, not all horizontal lines are found. In this report, every third line is found, as seen by the solid tracks left by the algorithm in Fig 5. Another point requiring discussion is a backup technique, found for example at program line 1054, to handle rough edges.

Rough edges occur whenever thresholding is done to provide a binary image. This arises because of noise, electronic and otherwise, in the video signal. As the scene changes from dark to bright, pixel intensities increase from 0 to 127, going through the threshold value of 63 at an intermediate gray level. In the presence of noise, however, the binary image will contain some bright dots beyond the true edge, and conversely there will be some black dots within the true body of the object. The width of the zone of uncertainty depends on how well the projector and camera are focused and how uniform the surfaces of the plates are.

In conventional image processing, edges can be cleaned up through use of a region growing algorithm followed by region shrinking. This would be computationally expensive. Instead of image processing, we use a backup technique which will be successful provided the stripe width is greater than twice the uncertain zone at the edges. Thus we go to the last bright pixel at the edge and then backup a distance equal to the width of the zone of uncertainty. After backing up, the search continues in a new direction. The backing up can be seen at the end of each search line in Fig 5, where the rough edges can also be seen.



(a) (b) (c)

Fig 5. Simulated data for a wide gap and considerable misfit. (a) Image of stripe of light as seen with binary vision. (b) Path followed by search algorithm. (c) Composite graph showing image (cross-hatched) and search path.

The program was written in BASIC and was run with Z-Basic which compiles and runs fast. The program leaves tracks (optional) on the monitor to show where it has searched except for searches across breaks such as seen in Fig 5. Cycle time is about 0.15 sec. This speed of execution would be adequate for many welding applications, though increased speed could be obtained through use of a faster computer or perhaps by programming in assembly language.

In the above, the algorithm accessed 1742 pixels in the image, an amount of data comparable to that processed by Kawahara and Matsui/7/ with their 32 X 32 pixel sensor.

EXPERIMENTAL RESULTS

Observations were made on samples of metal plates prepared for welding with bevelled edges, using the computer vision system described above. Since no welding was performed, this demonstration showed how a vision system would perform in the two pass mode where a separate measurement pass is made before welding is performed.

Two samples with different bevel angles were observed. In the first case, with a small angle B (see Fig 2), the stripe of light was very faint on the bevelled surfaces. A small reflection from the left bevelled surface onto the right one could be seen in normal vision, but not in binary vision. Fig 6 shows the result of applying the search algorithm. Since the bevelled surfaces are not detected, the boxes for the edges are placed over those for the shoulders. However, this is still a useable result since the joint root, gap, and misfit can be calculated knowing the thickness T, bevel angle B, and positions of the shoulders.

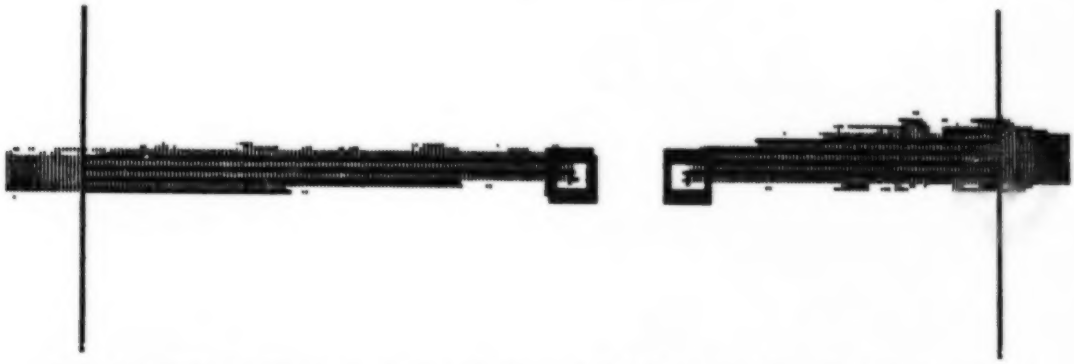


Fig 6. Narrow groove. The stripe of light was too faint in the groove area to see the edge. However, the shoulders of the groove were detected reliably so that the edges could be found by computation.

The second sample had a larger bevel angle so that the bevelled surface was closer to horizontal. The view that is seen by the camera given in Fig 7 shows the V-groove and some intrascene reflections. There is considerable difference in the intensity of light seen on the top surface from that seen across the groove. The top surface is in an "as received" condition and is to some extent a diffuse surface, which is oriented in a favourable direction to be seen. On the other hand, the groove walls are machined to a shiny appearance and are oriented at steep angles. Two effects follow from this. First, the radiance toward the camera is small. Second, there is a strong reflection across the groove to the other side.

To improve the optical properties of the sample a diffuse coating was applied. Two coatings were tried: white paper to simulate white matte paint, and soot particles from a smokey torch. Both had the desired effect of providing a good stripe of light on the bevelled surfaces, and eliminating the reflection across to the other side of the groove.

Fig 7. Wide groove with shiny walls. Although weaker, the stripe of light in the groove area could be detected sporadically using the same threshold as used for the top surface. The unexpected large bright areas are reflections from the opposite groove wall.

Successful operation of the computer vision system is seen in Fig 8, where the gap position, gap width, and a slight misfit are seen.

Fig 8. Coated sample can be tracked by search algorithm. Ideal image was obtained when sample was coated with soot to give a matte surface with no strong reflections.

DISCUSSION ON IMPROVEMENTS TO PROTOTYPE SYSTEM

Binary vision, which permits rapid computer processing, showed excellent results on the stripe of light on the top surface, but poor results in the groove. Furthermore, the secondary reflection was seen to be stronger than the desired stripe of light for the case of a steep walled groove, and about equal for a wider groove.

Several options could be considered to deal with the complexities described above. In one scheme, only the stripe of light on the top surface would be used, leaving it up to computation to locate the actual "weld root" from the measured shoulder positions on the top surface, and from the plate thickness and bevel angle. This option would unfortunately require that joint preparations be precision machined, thereby obviating one of the potential advantages of machine vision, namely the ability to compensate for machining variations. In a second method, a special coating would be applied to provide a matte surface, either white or black. While such coatings would mould the problem to suit a simple machine vision system, they would require additional steps in the welding process and could interfere with weld quality.

Alternatively, one could resort to a more sophisticated vision system than the one described in this report. A combination of binary and gray level vision could be used; for example, binary vision could be used as given in this report to find the shoulders of the weld groove, and then gray level vision or binary vision with a lower threshold could be used in the groove region to bring out the faint reflection from the groove walls. Another approach to equalize the stripe of light seen on the top surface with that seen from the groove walls would be optical; for example a Polaroid filter could be adjusted to take advantage of the different depolarization of polarized illumination expected from the cold-rolled top surface and the machined groove walls.

To deal with the unwanted secondary reflections, various software tests and limitations could be imposed. If required, the physics and optics of the diverse surface problem could be described by a mathematical model which would be fitted to the image in order to infer joint geometry.

While the personal computer (programmed in Z-Basic) used in this work was only just able to achieve the target for system speed even though the simplest possible program was used, new advances in faster computers which have recently become available with no increase in price would provide a speed margin for more sophisticated programs.

To provide positional data for the welding robot, the optical triangulation calculations indicated in Fig 4 would be incorporated to translate from video screen coordinates to spatial coordinates.

CONCLUSIONS

A computer vision system has been demonstrated for obtaining the gap centre, gap height, gap width, and misfit for a butt weld between two plates or pipe ends. Low cost off-the-shelf hardware was used for the illumination source, video camera, computer, and frame grabber (the interface between video camera and computer.) Software was written in BASIC and could be modified easily by any user to suit his particular problem. When compiled, execution time was fast enough (0.15 sec) for many applications.

Experience has been gained in some of the problems that are faced by any welding vision system (either in-house or external.) A fundamental problem is the reflection of light from one wall of the groove over to the other. Suggestions are given to deal with this problem.

ACKNOWLEDGEMENTS

Incentive to pursue this work was provided by a need for a sensor system in the automated welding research being done by the Welding Unit in the Metallurgical Research Department. Papers from the literature were provided by personnel in the Welding Unit, who also set up the apparatus used for the experimental results shown in this report, and provided helpful discussions.

N.F. Pereira pointed out the speed benefits of Z-Basic.

Approved:

A.F. Baljet
Manager
Electrical Research Department

Submitted:

D.L. Mader

D.L. Mader
Associate Research Scientist
Science Section

REFERENCES

1. Grant, I.M., Maak, P.Y.Y., and Tinkler, M.J., "An assessment of automated pipe welding - literature survey," Report No. 86-27-K, Research Division, Ontario Hydro, Toronto, Canada M8Z 5S4 (1986).

2. Drews, P. and Starke, G., "Development approaches for advanced adaptive control in automated arc welding," Proceedings of the Conference on Automation and Robotization in Welding and Allied Processes (held by the International Institute of Welding), Pergamon (1985).

3. Agapakis, J.E., Wittels, N. and Masubuchi, K., "Automated visual weld inspection for robotic welding fabrication," Proceedings of the Conference on Automation and Robotization in Welding and Allied Processes (held by the International Institute of Welding), Pergamon (1985).

4. Ozeki, O., Nakano, T. and Yamamoto, S., "Real-time range measurement device for three-dimensional object recognition," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, 550 (1986).

5. Ballard, Dana H. and Brown, Christopher M., Computer Vision, Prentice-Hall (1982).

6. Begin, G., Boillot, J.P., Michel, C. and Teubel, G., "Third generation adaptive robotic arc welding unit," Proceedings of the Conference on Automation and Robotization in Welding and Allied Processes (held by the International Institute of Welding), Pergamon (1985).

7. Kawahara, M. and Matsui, H., "Tracking control system for arc welding using image sensor," IPAC Control Science and Technology (8th Biennial World Congress), Kyoto, Japan, p.2117 (1981).

APPENDIX. COMPUTER PROGRAM

```
00001 CLS
00002 PRINT "                    PROGRAM ZSEAM"
00003 PRINT "              SEAM TRACKING ON OCULUS-200"
00004 PRINT "                FOR AUTOMATIC WELDING"
00005 PRINT "             Version for use with Z-Basic"
00006 PRINT "     DAVID L. MADER, ONTARIO HYDRO, TORONTO, CANADA"
00007 PRINT "                      AUGUST 12, 1986"
00008 PRINT "" : PRINT ""
00009 DIM JV(20),HIT%(512),HOSZ%(512),XSIGN%(2),XSTART%(2),XMAXHOSZ%(2)
00010 DIM ADVANCE%(2),RETREAT%(2),WALL%(2),X%(2,2),Y%(2,2) : REM  % is integer.
00011 REM*************************************************************************
00012 REM  This is the main program.
00013 REM*******************************
00014 GOSUB 100     : REM  Set up frame grabber.
00015 GOSUB 200     : REM  Set background/clear screen.
00016 GOSUB 300     : REM  Gray-level vision (to focus camera).
00017 GOSUB 400     : REM  Binary vision (to set aperture for correct contrast).
00018 GOSUB 500     : REM  Massage operating data.
00019 GOSUB 1000    : REM  Find feature points in image.
00020 INPUT"? Want to redo (Y) ?";C$
00021 IF C$ = "Y" OR C$="y" THEN GOTO 14
00022 INPUT"? Want to store weld vision operating data (Y) ?";C$
00023 IF C$ >< "Y" AND C$ >< "y" THEN END
00024 OPEN "O",#1, "WELD-VIS.PAR"
00025 FOR I=1 TO 20 : PRINT#1,JV(I) : NEXT I
00026 CLOSE#1
00027 END
00100 REM*************************************************************************
00101 REM     Set up frame grabber board.
00102 REM***********************************
00103 REM Frame grabber is Oculus-200 made by CORECO INC., 555 St-Thomas St.,
00104 REM   Longueuil, Quebec, Canada J4H 3A7.   Phone (514) 651-3100.
00105 REM Borders of screen. Oculus screen Left, Top, Right, Bottom
00106 OCL=0 : OCT=1 : OCR=511 : OCB=482 : JT%=OCT : JB%=OCB
00107 REM  I/O ports.    OUT and INP are for machine ports.
00108       DAT       = &HF10    : REM  Data register.
00109       DIR       = &HF11    : REM  Direction register.
00110       STATUS    = &HF15    : REM  Status register.
00111       XREGH     = &HF11    : XREGL        = &HF12
00112       YREGH     = &HF13    : YREGL        = &HF14
00113 REM  Directions for auto-incrementing:
00114 NOSTH%=10:SOVTH%=8:EAST%=4:WEST%=5:REM Strange spelling to avoid key words.
00115 REM  Set numerical values of Direction register command bits.
00116   GRAFACC%=16 : REM  Graphics plane access = /WRPRT7.
00117   LUTACC%=32  : REM  Look Up Table access.
00118   FD%=64      : REM  Frame grab from Data register. Enable DMA channel.
00119   RMW%=128    : REM  Inhibit auto-increment on Read in Read-Modify-Write.
00120 REM***************************************
```

```
00121 REM  Set input lookup tables.
00122 REM*****************************
00123 OUT XREGH,0 : OUT YREGH,0
00124 REM *** Binary Vision ***
00125 REM  3 bits for LUT number: X0,X1,D0(UPX).
00126 REM  To load, set "LUTACC" (LUT access), D5=1.
00127 OUT XREGL,0 : REM  Select input LUT number 0.
00128 OUT YREGL,0 : REM  Starting point for Y to auto-increment from.
00129 OUT DIR,LUTACC%+SOVTH%      : REM  LUT access and auto-increment South.
00130 REM Binary vision threshold 63. While grabbing, adjust camera/lighting.
00131 FOR I=0 TO 127
00132 IF I < 63 THEN OUT DAT,0
00133 IF I >= 63 THEN OUT DAT,127
00134 NEXT I
00135 REM *** Normal gray-level vision ***
00136 REM  Set an input LUT at X=1 with unity transformation, to focus camera.
00137 OUT XREGL,1 : REM  Select input LUT number 1.
00138 OUT YREGL,0 : REM  Starting point for Y to auto-increment from.
00139 FOR I=0 TO 127
00140 OUT DAT,I
00141 NEXT I
00142 RETURN
00200 REM*************************************************************************
00201 REM   Set screen background to intensity K ( 0 for clear screen ).
00202 REM*************************************************************************
00203 REM  Same as frame grab, except Direction register bit 6 set to 1. (FD=1)
00204 REM  To erase graphics plane, set "       "     bit 4 to 1 (/WRPRT7=1).
00205 INPUT"? Want to clear Oculus screen (Y) ?";C$
00206 IF C$ >< "Y" AND C$ >< "y" THEN RETURN
00207 INPUT"? Background intensity ?";K
00208 OUT DAT,K
00209 OUT DIR,FD%+GRAFACC%       : REM  Set FD=1 and /WRPRT7=1.
00210 GOSUB 518                  : REM  Frame grab.
00211 RETURN
00300 REM*************************************************************************
00301 REM  Gray-level vision to adjust camera or lighting.
00302 REM*************************************************
00303 INPUT"? Want gray-level vision (Y) ?";C$
00304 IF C$ >< "Y" AND C$ >< "y" THEN RETURN
00305 OUT DIR,0                          : REM  Load mode.
00306 OUT XREGL,1                        : REM  Select normal vision LUT.
00307 PRINT" Press Enter when done."
00308 OUT DIR,GRAFACC%
00309 GOSUB 518                          : REM  Grab image.
00310 C$=INKEY$
00311 IF C$ >< CHR$(13) THEN GOTO 309 : REM Keep on grabbing until see an Enter
00312 RETURN
00400 REM*************************************************************************
00401 REM  Binary vision to adjust camera aperture.
00402 REM*********************************************
00403 INPUT"? Want binary vision (Y) ?";C$
```

```
00404 IF C$ >< "Y" AND C$ >< "y" THEN RETURN
00405 PRINT" Adjust camera. Press Enter when done."
00406 OUT DIR,0          : REM  Load mode.
00407 OUT XREGL,0        : REM  Select binary vision LUT.
00408 OUT DIR,GRAFACC%
00409 GOSUB 518          : REM  Frame grab.
00410 C$=INKEY$
00411 IF C$ >< CHR$(13) THEN GOTO 409 : REM Keep on grabbing until see an Enter
00412 RETURN
00500 REM*****************************************************************************
00501 REM    Massage operating data.
00502 REM*****************************
00503 OPEN "I",#1,"WELD-VIS.PAR"
00504 FOR I=1 TO 20 : INPUT#1,JV(I) : NEXT I
00505 CLOSE#1
00506 PRINT"1   MS        ";JV(1)
00507 PRINT"2   BACKUP    ";JV(2)
00508 PRINT"3   ILSTART   ";JV(3)
00509 PRINT"4   IRSTART   ";JV(4)
00510 PRINT"5   BOX SIZE  ";JV(5)
00511 PRINT"6   CONVERGE  ";JV(6)
00512 PRINT"7   SEARCH    ";JV(7)
00513 PRINT"8   MAXCOUNT  ";JV(8)
00514 PRINT"9   TRACKS    ";JV(9)
00515 INPUT"? Changes   I,JV(I)  (Done?  0,0) ?";I,JV(I)
00516 IF I=0 THEN RETURN
00517 GOTO 515
00518 REM*****************************************************************************
00519 REM   Grab an image.
00520 REM**********************
00521 REM  Using "STATUS" register.
00522 REM  BIT S5 - Frame grab in progress = Busy.
00523 REM  BIT S7 - Vertical sync.  S6 - Even/Odd field ( 2 fields = frame ).
00524 REM  We have selected look-up table and direction register already.
00525 REM  LUTACC,D5=0 for grab. Set WRTPR7=1 to erase power-up graphics plane.
00526 REM  This subroutine synchronizes the computer to the frame grabber.
00527 VS=INP(STATUS) AND 128
00528 IF VS=0 THEN GOTO 527      : REM  Display may be in progress.
00529 VS=INP(STATUS) AND 128
00530 IF VS=128 THEN GOTO 529    : REM  Wait out blanking interval or part of it
00531 VS=INP(STATUS) AND 128
00532 IF VS=0 THEN GOTO 531      : REM  Wait out full half frame.
00533 REM  If we are here, we have detected the low to high transition of VS.
00534 REM  We are at the top of either the even or the odd half frame.
00535 REM  Writing into the Status register produces a frame grab signal.
00536 OUT STATUS,0
00537 REM  Do another grab for other half frame.
00538 FG=INP(STATUS) AND 32 : REM  Frame grab in progress when STATUS BIT 5 =1.
00539 IF FG=32 THEN GOTO 538
00540 OUT STATUS,0           : REM  Grab next half frame.
00541 FG=INP(STATUS) AND 32  : REM  Wait until second half completely grabbed.
```

```
00542 IF FG=32 THEN GOTO 541
00543 RETURN
00544 REM*************************************************************************
00545 REM       Draw box on Oculus screen.
00546 REM************************************
00547 REM auto-increment mode means IR and JR are incremented
00548 REM       at each "OUT DAT,---".
00549 REM Bits B2,B3 in DIR are:   non-zero for auto-increment
00550 REM                         zero for stationary.
00551 BOYX%=JV(5)
00552 FOR ILL=1 TO 2
00553 OUT DIR,(256 AND (J%+BOYX%+ILL))/128:OUT YREGL,255 AND (J%+BOYX%+ILL)
00554 OUT DIR,(256 AND (I%-BOYX%-ILL))/256:OUT XREGL,255 AND (I%-BOYX%-ILL)
00555 OUT DIR,GRAFACC%+NOSTH% : FOR IL=1 TO 2*BOYX%+2*ILL: OUT DAT,255 : NEXT
00556 OUT DIR,GRAFACC%+EAST% : FOR IL=1 TO 2*BOYX%+2*ILL : OUT DAT,255 : NEXT
00557 OUT DIR,GRAFACC%+SOVTH% : FOR IL=1 TO 2*BOYX%+2*ILL : OUT DAT,255 : NEXT
00558 OUT DIR,GRAFACC%+WEST% : FOR IL=1 TO 2*BOYX%+2*ILL : OUT DAT,255 : NEXT
00559 NEXT ILL
00560 REM  The loop above draws on both even and odd frames to avoid flicker.
00561 RETURN
01000 REM*************************************************************************
01001 REM   Find Feature Points on stripes of light.
01002 REM*********************************************
01003 PRINT"" :   PRINT"Press any key to quit tracking."   : PRINT""
01004 PRINT "Left shoulder  Left foot  Right shoulder   Right foot"
01005 TRACKS%=JV(9) : REM  Set to 0 for no tracks left where searched.
01006 MS%=JV(1) : BACKUP%=JV(2) : COMVERGE%=JV(6):SEARCH%=JV(7):MAXCOUNT%=JV(8)
01007 XSIGN%(1)=1 : XSIGN%(2)=-1 : XSTART%(1)=JV(3) : XSTART%(2)=JV(4)
01008 XMAXHOSZ%(1)=0 : XMAXHOSZ%(2)=512
01009 ADVANCE%(1)=EAST%: ADVANCE%(2)=WEST%: RETREAT%(1)=WEST%:RETREAT%(2)=EAST%
01010 WALL%(1)=JV(4) : WALL%(2)=JV(3)
01011 DO        : REM  A loop to line 1098, starting with a binary frame grab.
01012 OUT DIR,0 : OUT XREGL,0 : OUT DIR,GRAFACC%
01013 GOSUB 518 : REM  Frame grab.
01014 COUNT%=0  : REM  To put limit on time spent on one frame. (Unusable ones)
01015 REM*******   First, left line with SIDE=1 and then right one   **********
01016 REM   Do each side separately or lose speed benefit of auto-incrementing.
01017 FOR SIDE%=1 TO 2
01018 I%=XSTART%(SIDE%) : SIGN%=XSIGN%(SIDE%)
01019 OUT DIR,(256 AND (JT%+1))/128:OUT YREGL,255 AND (JT%+1)
01020 OUT DIR,(256 AND I%)/256:OUT XREGL,255 AND I%    : REM  Load mode.
01021 OUT DIR,GRAFACC%+SOVTH%                          : REM  Auto-increment south.
01022 IF TRACKS% >< 0 THEN OUT DIR,RMW%+GRAFACC%+SOVTH%
01023 REM   Do scan down each side and find all bright dots. Store in "HIT".
01024 FOR J%=JT%+1 TO JB%-1
01025   HIT%(J%)=0 : HOSZ%(J%)=I%
01026   IMTENSITY%=INP(DAT) : IF TRACKS% >< 0 THEN OUT DAT,255
01027   COUNT%=COUNT%+1 : IF COUNT% > MAXCOUNT% THEN GOTO 1098
01028   IF IMTENSITY%=0 THEN GOTO 1030
01029   HIT%(J%)=1
01030 NEXT J%
```

```
01031 REM Now, for all dots found above, find length of horizontal bright line.
01032 FOR J%=JT%+1 TO JB%-1 STEP MS%        : REM  Scan down side.
01033   IF HIT%(J%)=0 THEN GOTO 1046        : REM  Skip to bottom.
01034   OUT DIR,(256 AND J%)/128:OUT YREGL,255 AND J%
01035   OUT DIR,(256 AND I%)/256:OUT XREGL,255 AND I%
01036   OUT DIR,GRAFACC%+ADVANCE%(SIDE%)
01037   IF TRACKS% >< 0 THEN OUT DIR,RMW%+GRAFACC%+ADVANCE%(SIDE%)
01038   REM  Each time we do "INP(DAT)", X is incremented automatically.
01039   IMTENSITY%=INP(DAT)
01040   COUNT%=COUNT%+1 : IF COUNT% > MAXCOUNT% THEN GOTO 1098
01041   IF IMTENSITY%=0 THEN GOTO 1046
01042   IF TRACKS% >< 0 THEN OUT DAT,255
01043   HOSZ%(J%)=HOSZ%(J%)+SIGN%
01044   IF SIGN%*HOSZ%(J%) > SIGN%*WALL%(SIDE%) THEN GOTO 1098
01045   GOTO 1039
01046 NEXT J%
01047 REM  Now sort through "LENGTH" to find the longest line.
01048 MAXHOSZ%=XMAXHOSZ%(SIDE%) : JMAX%=JT%+1
01049 FOR J%=JT%+1 TO JB%-1 STEP MS%
01050   IF SIGN%*HOSZ%(J%) < SIGN%*MAXHOSZ% THEN GOTO 1052
01051   MAXHOSZ%=HOSZ%(J%) : JMAX%=J%
01052 NEXT J%
01053 J%=JMAX%
01054 I%=HOSZ%(J%)-SIGN%*BACKUP% : REM  Go to end of longest line and back up.
01055 REM  Draw box at shoulder (top surface to weld preparation).
01056 IF TRACKS% >< 0 THEN GOSUB 544
01057 X%(1,SIDE%)=I% : Y%(1,SIDE%)=J%
01058 OUT DIR,(256 AND J%)/128:OUT YREGL,255 AND J%
01059 OUT DIR,(256 AND I%)/256:OUT XREGL,255 AND I%
01060 REM*******  Do staircase search  *****************
01061 IO%=I% : JO%=J%
01062 OUT DIR,GRAFACC%+SOVTH%     : REM  Travel down and find bottom of stripe.
01063 IF TRACKS% >< 0 THEN OUT DIR,RMW%+GRAFACC%+SOVTH%
01064 IMTENSITY%=INP(DAT) : IF TRACKS% >< 0 THEN OUT DAT,255
01065 COUNT%=COUNT%+1 : IF COUNT% > MAXCOUNT% THEN GOTO 1098
01066 J%=J%+1 : IF J% >= JB% THEN GOTO 1098
01067 IF IMTENSITY% > 63 THEN GOTO 1064
01068 REM  If we are here, we have run off bottom of stripe.
01069 OUT DIR,NOSTH%
01070 FOR K%=1 TO BACKUP%          : REM  Backup a little.
01071   J%=J%-1 : CAT%=INP(DAT)
01072 NEXT K%
01073 OUT DIR,GRAFACC%+ADVANCE%(SIDE%)          : REM  Travel across for side.
01074 IF TRACKS% >< 0 THEN OUT DIR,RMW%+GRAFACC%+ADVANCE%(SIDE%)
01075 IMTENSITY%=INP(DAT):IF TRACKS% >< 0 THEN OUT DAT,255
01076 COUNT%=COUNT%+1 : IF COUNT% > MAXCOUNT% THEN GOTO 1098
01077 I%=I%+SIGN%:IF I% >= JV(4) THEN GOTO 1098
01078 IF IMTENSITY% > 63 THEN GOTO 1075
01079 REM  Have run off side of stripe or hit a break.
01080 OUT DIR,RETREAT%(SIDE%)
01081 FOR K%=1 TO BACKUP%          : REM  Backup a little.
```

```
01082 I%=I%-SIGN% : CAT%=INP(DAT)
01083 NEXT K%
01084 IF ABS(I%-IO%)+ABS(J%-JO%) > COMVERGE%*BACKUP% THEN GOTO 1061 :REM  Again
01085 REM  Search a bit farther along to see if the stripe resumes.
01086 OUT DIR,ADVANCE%(SIDE%)
01087 FOR K%=1 TO SEARCH%
01088 I%=I%+SIGN%
01089 COUNT%=COUNT%+1 : IF COUNT% > MAXCOUNT% THEN GOTO 1098
01090 IF INP(DAT)=127 THEN GOTO 1061     : REM  Found continuation of stripe.
01091 NEXT K%
01092 I%=I%-SIGN%*SEARCH%
01093 X%(2,SIDE%)=I% : Y%(2,SIDE%)=J%    : REM  Second box, at foot.
01094 REM  Plot box at end of line on Oculus screen.
01095 IF TRACKS% >< 0 THEN GOSUB 544
01096 NEXT SIDE%
01097 PRINT X%(1,1);"  ";Y%(1,1);"  ";X%(2,1);"  ";Y%(2,1);"  ";
           X%(1,2);"  ";Y%(1,2);"  ";X%(2,2);"  ";Y%(2,2)
01098 UNTIL LEN(INKEY$) : REM Keep going until any key is pressed.
01099 RETURN
```

This software was created using the ZBasic™ Compiler.
Portions of this Code are © Copyrighted, 1985 by Zedcor, Inc.